

Yet another fast variant of Newton’s method for nonconvex optimization

S. Gratton*, S. Jerad† and Ph. L. Toint‡

20 II 2023

Abstract

A second-order algorithm is proposed for minimizing smooth nonconvex functions that alternates between regularized Newton and negative curvature steps. In most cases, the Hessian matrix is regularized with the square root of the current gradient and an additional term taking moderate negative curvature into account, a negative curvature step being taken only exceptionnally. As a consequence, the proposed method only requires the solution of a single linear system at nearly all iterations. We establish that at most $\mathcal{O}(|\log \epsilon| \epsilon^{-3/2})$ evaluations of the problem’s objective function and derivatives are needed for this algorithm to obtain an ϵ -approximate first-order minimizer, and at most $\mathcal{O}(|\log \epsilon| \epsilon^{-3})$ to obtain a second-order one. Initial numerical experiments with two variants of the new method are finally presented.

Keywords: Newton’s method, nonconvex optimization, negative curvature, adaptive regularization methods, evaluation complexity.

1 Introduction

It is not an understatement to say that Newton’s method is a central algorithm to solve nonlinear minimization problems, mostly because the method exhibits a quadratic rate of convergence when close to the solution and is affine-invariant. In the worst case, it can however be as slow as a vanilla first-order method [7], [12, Section 3.2] even when globalized with a linesearch [35] or a trust region [14]. This drawback has however been circumvented by the cubic regularization algorithm [36] and its subsequent adaptive variants [8, 9], [12, Section 3.3]. For nonconvex optimization, these latter variants exhibit a worst-case $\mathcal{O}(\epsilon^{-3/2})$ complexity order to find an ϵ - first-order minimizer compared with the $\mathcal{O}(\epsilon^{-2})$ order of second-order trust-region methods [26], [12, Section 3.2]. Adaptive cubic regularization was later extended to handle inexact derivatives [40, 41, 2, 1], probabilistic models [1, 13], and even schemes in which the value of the objective function is never computed [24].

However, as noted in [33], the improvement in complexity has been obtained by trading the simple Newton step requiring only the solution of a single linear system for more complex or

*Université de Toulouse, INP, IRIT, Toulouse, France. Email: serge.gratton@enseeiht.fr. Work partially supported by 3IA Artificial and Natural Intelligence Toulouse Institute (ANITI), French ”Investing for the Future - PIA3” program under the Grant agreement ANR-19-PI3A-0004”

†ANITI, Université de Toulouse, INP, IRIT, Toulouse, France. Email: sadok.jerad@toulouse-inp.fr

‡NAXYS, University of Namur, Namur, Belgium. Email: philippe.toint@unamur.be. Partly supported by ANITI.

slower procedures, such as secular iterations, possibly using Lanczos preprocessing [6, 8] (see also [12, Chapters 8 to 10]) or (conjugate-)gradient descent [29, 4]. In the simpler context of convex problems, two recent papers [33, 17] independently proposed another globalization technique. At an iterate x , the step s is computed as

$$s = -(\nabla_x^2 f(x) + \lambda_k I_n)^{-1} \nabla_x^1 f(x) \quad (1.1)$$

where $\lambda_k \sim \sqrt{\|\nabla_x^1 f(x)\|}$. This new approach exhibits the best complexity rate of second-order methods for convex optimization and retains the local superlinear convergence of standard Newton method, while showing remarkable numerical promise [33]. Devising an algorithm for nonconvex functions that can use similar ideas whenever possible appears as a natural extension.

In the nonconvex case, the Hessian may be indefinite and it is well-known that negative curvature can be exploited to ensure progress towards second-order points. Mixing gradient-related (possibly Newton) and negative curvature directions has long been considered and can be traced back to [32], which initiated a line of work using curvilinear search to find a step combining both types of directions. The length of the step is typically tuned using an Armijo-like condition [32, 20, 34]. Improvements were subsequently proposed by incorporating the curvilinear step in a nonmonotone algorithm [19], allowing the resolution of large-scale problems [31] or by choosing between the two steps based on model decrease [22]. Alternatively, negative curvature has also been used to regularize the Hessian matrix, yielding the famous Goldfeld-Quandt-Trotter (GQT) method [21]. Unfortunately, this method also involves more complex computation to find the step and has the same global convergence rate as first-order algorithms [39].

One may then wonder if it is possible to devise an adaptive second-order method using a single explicitly regularized Newton step when possible and a negative curvature direction only when necessary, with a near-optimal complexity rate. The objective of this paper is to show that it is indeed possible (and efficient). To this aim, we propose a fast Newton's method that exploits negative curvature for nonconvex optimization problems and generalizes the method proposed in [33, 17] to the nonconvex case. The new algorithm automatically adjusts the regularization parameter (without knowledge of the Hessian's Lipschitz constant). It first attempts a step along a direction regularized only by the square root of the gradient only, as in the convex setting [17, 33]. In that sense, it is inspired by the "convex until proved guilty" strategy advocated by [5]. If this attempt fails, the method either uses an appropriately regularized Newton step taking the smallest negative eigenvalue of the Hessian also into account or simply follows the negative curvature otherwise. We prove that this method requires at most $\mathcal{O}(|\log \epsilon| \epsilon^{-3/2})$ iterations and evaluations of the problem data to obtain an ϵ -approximate first-order critical point, which is very close to the optimal convergence rate of second-order methods for Lipschitz Hessian functions [10]. We also introduce an algorithmic variant of the new method which is guaranteed to find a second-order critical point in at most $\mathcal{O}(|\log \epsilon| \epsilon^{-3})$ iterations.

The paper is organized as follows. Section 2 describes the algorithm and one of its variants and compares it with recent work on second-order methods. Section 3 states our assumptions and derives a bound on its worst-case complexity for finding first-order critical points, while Section 4 presents the second-order algorithmic variant and analyzes its complexity. Section 5 then illustrates the numerical behaviour of the proposed methods. Some conclusions are finally drawn in Section 6.

Notation The following notations will be used throughout the paper. The symbol $\|\cdot\|$ denotes the Euclidean norm for vectors in \mathbb{R}^n and its associated subordinate norm for matrices. $\lambda_{\min}(M)$ denotes the minimum eigenvalue of a symmetric matrix M , while I_n is the identity matrix in \mathbb{R}^n . For a real $x \in \mathbb{R}$, we define $[x]_+ = \max(x, 0)$. For two vectors $x, y \in \mathbb{R}^n$, we denote $x^\top y$ their scalar product.

2 Adaptive Newton with Negative Curvature

We consider the problem of finding approximate first-order critical points of the smooth unconstrained nonconvex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2.1)$$

and discuss our algorithm called AN2C (for Adaptive Newton with Negative Curvature) on the following page. The algorithm, whose purpose is to compute first-order critical points, is presented in the framework of adaptive regularization methods [3, 8] [12, Section 3.3] and proceeds as follows. At each iteration, a regularized Newton step is first attempted using a regularization based on first-order information only. If the associated matrix is indefinite, a more elaborate second-order step is computed using the `EigenNewtonStep` subroutine. In this case, a (more strongly) regularized Newton step is computed when the negative curvature is less than a fraction of the square root of the product of the gradient norm with the regularization parameter. If this latter condition fails, the algorithm uses a step along the negative curvature direction to compute the trial step. The details are given in Algorithms 2.1 and 2.2, on the next page and on page 5, respectively.

The s_k^{conv} notation in (2.2) stresses the connection with the regularization technique proposed in the convex framework [33, 17]. Indeed, the system (2.2) is well posed in a region where the Hessian is positive definite and condition (2.4) is then always satisfied. We emphasize that checking conditions (2.3) to (2.5) is possible and cheap compared to the computation of the smallest eigenvalue in the `EigenNewtonStep` procedure. This is in particular the case for (2.4) since a Cholesky factorization delivers the necessary information. The test (2.5) is required as to avoid steps whose magnitude is too large compared to the gradient (the motivation for its particular form of the test will become clear in Section 3).

When computing a vector s_k^{conv} satisfying (2.3) to (2.5) is not possible, a call to the `EigenNewtonStep` procedure is made. In this procedure, the notation s_k^{neig} (2.9) refers to the combined use of a regularized Newton and negative curvature information. Having computed $[-\lambda_{\min}(H_k)]_+$, the Hessian can then be sufficiently regularized to ensure the existence and the descent nature of the step resulting from (2.9). Note that we do not impose an exact solution of the linear system but allow termination of the linear algebra solver provided condition (2.10) holds. As it turns out, $[-\lambda_{\min}(H_k)]_+$ could have been replaced by $\kappa_C \sqrt{\sigma_k \|g_k\|}$ in (2.9) and the remainder of the complexity analysis would remain valid. An interesting connection can be established between the regularization in (2.9) and the GQT method [21], as the regularization parameter $(\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+)$ is very similar in spirit to that used in this method. Also observe that, in most cases, the ‘‘approximate minimum curvature

Algorithm 2.1: Adaptive Newton with Negative Curvature (AN2C)

Step 0: Initialization: An initial point $x_0 \in \mathbb{R}^n$, a regularization parameter $\sigma_0 > 0$ and a gradient accuracy threshold $\epsilon \in (0, 1]$ are given, as well as the parameters

$$\sigma_{\min} > 0, \kappa_C, \kappa_\theta > 0, \kappa_a \geq 1, \varsigma_1 \in (0, 1), \varsigma_2 \in [0, \frac{1}{2}), \varsigma_3 \in [0, 1),$$

$$0 < \gamma_1 < 1 < \gamma_2 \leq \gamma_3 \quad \text{and} \quad 0 < \eta_1 \leq \eta_2 < 1,$$

Set $k = 0$.

Step 1: Compute current derivatives Evaluate $g_k \stackrel{\text{def}}{=} \nabla_x^1 f(x_k)$ and $H_k \stackrel{\text{def}}{=} \nabla_x^2 f(x_k)$. Terminate if

$$\|g_k\| \leq \epsilon.$$

Step 2: Step calculation Attempt to solve the linear system

$$(H_k + \sqrt{\kappa_a \sigma_k \|g_k\|} I_n) s_k^{\text{conv}} = -g_k \quad (2.2)$$

such that the system residual $r_k = (H_k + \sqrt{\kappa_a \sigma_k \|g_k\|} I_n) s_k^{\text{conv}} + g_k$ satisfies

$$\|r_k^{\text{conv}}\| \leq \min \left(\varsigma_2 \sqrt{\kappa_a \sigma_k \|g_k\|} \|s_k^{\text{conv}}\|, \kappa_\theta \|g_k\| \right). \quad (2.3)$$

and

$$H_k + \sqrt{\kappa_a \sigma_k \|g_k\|} I_n \succ 0. \quad (2.4)$$

If such an s_k^{conv} can be computed and

$$\|s_k^{\text{conv}}\| \leq \frac{(1 + \kappa_\theta)}{\varsigma_1} \sqrt{\frac{\|g_k\|}{\kappa_a \sigma_k}}; \quad (2.5)$$

set $s_k = s_k^{\text{conv}}$. Else define s_k by

$$s_k = \text{EigenNewtonStep}(g_k, H_k, \sigma_k, \kappa_C, \kappa_\theta, \varsigma_3). \quad (2.6)$$

Step 3: Acceptance ratio computation Evaluate $f(x_k + s_k)$ and compute the acceptance ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{-(g_k^\top s_k + \frac{1}{2} s_k^\top H_k s_k)}. \quad (2.7)$$

If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$ else $x_{k+1} = x_k$.

Step 4: Regularization parameter update Set

$$\sigma_{k+1} \in \begin{cases} [\max(\sigma_{\min}, \gamma_1 \sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_2 \sigma_k, \gamma_3 \sigma_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (2.8)$$

Increment k by one and go to Step 1.

Algorithm 2.2: EigenNewtonStep($g_k, H_k, \sigma_k, \kappa_C, \kappa_\theta, \varsigma_3$)

Compute $\lambda_{\min}(H_k)$.

If $-\lambda_{\min}(H_k) \leq \kappa_C \sqrt{\sigma_k \|g_k\|}$,

approximately solve the following linear system

$$\left(H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+) I_n \right) s_k^{neig} = -g_k. \quad (2.9)$$

to ensure the residual condition

$$\begin{aligned} \|r_k^{neig}\| &\stackrel{\text{def}}{=} \left\| \left(H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+) I_n \right) s_k^{neig} + g_k \right\| \\ &\leq \min \left(\varsigma_3 \sqrt{\sigma_k \|g_k\|} \|s_k^{neig}\|, \kappa_\theta \|g_k\| \right) \end{aligned} \quad (2.10)$$

and set $s_k = s_k^{neig}$.

If $-\lambda_{\min}(H_k) > \kappa_C \sqrt{\sigma_k \|g_k\|}$,

compute v_k such that

$$g_k^\top v_k \leq 0, \quad \|v_k\| = 1 \quad \text{and} \quad v_k^\top H v_k \leq -\kappa_C \sqrt{\sigma_k \|g_k\|}, \quad (2.11)$$

and set

$$s_k = s_k^{curv} = \frac{\kappa_C \sqrt{\sigma_k \|g_k\|}}{\sigma_k} v_k. \quad (2.12)$$

Return s_k .

direction” v_k is already available when computing $\lambda_{\min}(H_k)$. It can be also retrieved via a Lanczos procedure as proposed in [38, Lemma 9].

Once the step has been computed, the mechanisms of the algorithm for accepting/rejecting the new iterate (Step 3) and updating the regularization parameter (Step 4) are typical of adaptive regularization algorithms (see [12, Section 3.3.1], for instance).

An important feature of the AN2C algorithm is that, if a direct solver is used for (2.2) and (2.9), the process to compute the step s_k does not involve any inner iterative method beyond the (typically rare) computation of $[-\lambda_{\min}(H_k)]_+$, and only requires at most (and hopefully just one) linear-system solve.

We finally note that the computation involved in (2.2) to (2.5) are introduced solely in order to limit the need of possibly costly second-order information. Exploiting this feature, we also define a (potentially more costly) variant of AN2C, which we call AN2E, where no attempt is made in Step 2 to solve the system (2.2) and to verify (2.4), but where (2.6) is used at every iteration.

Before analyzing the complexity analysis of the AN2C method, we further explore its basic properties and discuss its relations with closely related nonconvex optimization algorithms. The method proposed in [15] differs from AN2C as it first takes a gradient step followed by a negative curvature one. [30] checks a condition before choosing between a gradient descent and a negative curvature direction. Note that this condition requires the knowledge of the smoothness parameters whereas AN2C is fully adaptive. Another related method is that of [38]. At variance with AN2C, this latter algorithm checks various conditions before choosing a specific direction (gradient, Newton, negative curvature) and performs a linesearch. [16] proposes a trust-region algorithm (at variance with adaptive regularization) that solves the trust-region subproblem by a combination of conjugate gradients and negative curvature. Note that their condition on the residuals of this subproblem [16, Inequality (27)] can be related to both (2.3) and (2.10).

Following well-established practice, we now define

$$\mathcal{S} \stackrel{\text{def}}{=} \{k \geq 0 \mid x_{k+1} = x_k + s_k\} = \{k \geq 0 \mid \rho_k \geq \eta_1\},$$

the set of indexes of “successful iterations”, and

$$\mathcal{S}_k \stackrel{\text{def}}{=} \mathcal{S} \cap \{0, \dots, k\},$$

the set of indexes of successful iterations up to iteration k . We further partition \mathcal{S}_k in three subsets depending on the nature of the step taken, so that

$$\mathcal{S}_k^{\text{neig}} \stackrel{\text{def}}{=} \mathcal{S}_k \cap \{s_k = s_k^{\text{neig}}\}, \quad \mathcal{S}_k^{\text{curv}} \stackrel{\text{def}}{=} \mathcal{S}_k \cap \{s_k = s_k^{\text{curv}}\}, \quad \mathcal{S}_k^{\text{conv}} \stackrel{\text{def}}{=} \mathcal{S}_k \cap \{s_k = s_k^{\text{conv}}\}.$$

We also recall a well-known result bounding the total number of iterations of adaptive regularization methods in terms of the number of successful ones.

Lemma 2.1 [3, Theorem 2.4],[12, Lemma 2.4.1] Suppose that the AN2C algorithm is used and that $\sigma_k \leq \sigma_{\max}$ for some $\sigma_{\max} > 0$. Then

$$k \leq |\mathcal{S}_k| \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{\max}}{\sigma_0} \right). \quad (2.13)$$

This result implies that the overall complexity of the algorithm can be estimated once bounds on σ_k and $|\mathcal{S}_k|$ are known, as we will show in the next section.

We now state a simple relation between $\|s_k^{neig}\|$, σ_k and $\|g_k\|$ inspired by [33].

Lemma 2.2 For all iterations k where s_k^{neig} is computed, we have that

$$g_k = - \left(H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+) I_n \right) s_k^{neig} + r_k^{neig} \quad (2.14)$$

and

$$\|s_k^{neig}\| \leq (1 + \kappa_\theta) \sqrt{\frac{\|g_k\|}{\sigma_k}}. \quad (2.15)$$

Similarly, when s_k^{conv} is computed,

$$g_k = - \left(H_k + \sqrt{\kappa_a \sigma_k \|g_k\|} I_n \right) s_k^{conv} + r_k^{conv}. \quad (2.16)$$

Proof. Equation (2.14) results from (2.9) and the definition of the residual. The (2.15) bound is deduced from (2.14), the fact that $H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+) I_n$ is a positive definite matrix with

$$\lambda_{\min}(H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+) I_n) \geq \sqrt{\sigma_k \|g_k\|}$$

and that $\|r_k^{neig}\| \leq \kappa_\theta \|g_k\|$ because of (2.10). If $k \in \mathcal{S}_k^{conv}$, (2.16) is obtained from (2.2) and the definition of the residual r_k^{conv} . \square

The next lemma gives a lower bound on the decrease of the local quadratic approximation. In standard adaptive regularization algorithms, this decrease automatically results from the minimization of the model (See [3] for instance). In our case, we need to use the properties of s_k^{conv} , s_k^{curv} and s_k^{neig} to obtain the desired result.

Lemma 2.3 Let k be a successful an iteration of AN2C. If $k \in \mathcal{S}_k^{conv}$, we have that

$$- \left(g_k^\top s_k + \frac{1}{2} s_k^\top H_k s_k \right) \geq \frac{1 - 2\varsigma_2}{2} \sqrt{\kappa_a \sigma_k \|g_k\|} \|s_k\|^2. \quad (2.17)$$

If $k \in \mathcal{S}_k^{neig}$, then

$$- \left(g_k^\top s_k + \frac{1}{2} s_k^\top H_k s_k \right) \geq (1 - \varsigma_3) \sqrt{\sigma_k \|g_k\|} \|s_k\|^2. \quad (2.18)$$

Else, if $k \in \mathcal{S}_k^{curv}$,

$$- \left(g_k^\top s_k + \frac{1}{2} s_k^\top H_k s_k \right) \geq \frac{1}{2} \sigma_k \|s_k\|^3. \quad (2.19)$$

Proof. Suppose first that $k \in \mathcal{S}_k^{conv}$. We then obtain from (2.16) that

$$\begin{aligned}
 g_k^\top s_k^{conv} + \frac{1}{2}(s_k^{conv})^\top H_k s_k^{conv} &\leq (r_k^{conv})^\top s_k^{conv} - (s_k^{conv})^\top (H_k + \sqrt{\kappa_a \sigma_k} \|g_k\| I_n) s_k^{conv} \\
 &\quad + \frac{1}{2}(s_k^{conv})^\top H_k s_k^{conv} \\
 &\leq -\sqrt{\kappa_a \sigma_k} \|g_k\| \|s_k^{conv}\|^2 + (r_k^{conv})^\top s_k^{conv} - \frac{1}{2}(s_k^{conv})^\top H_k s_k^{conv} \\
 &\leq -\sqrt{\kappa_a \sigma_k} \|g_k\| \|s_k^{conv}\|^2 + \varsigma_2 \sqrt{\kappa_a \sigma_k} \|g_k\| \|s_k^{conv}\|^2 \\
 &\quad + \frac{1}{2} \sqrt{\kappa_a \sigma_k} \|g_k\| \|s_k^{conv}\|^2.
 \end{aligned}$$

Hence (2.17) follows by using (2.4), (2.3) and that $\varsigma_2 \in [0, \frac{1}{2}]$ in the last line.

Suppose now that $k \in \mathcal{S}_k^{neig}$. By using (2.14) and the fact that $H_k + [-\lambda_{\min}(H_k)]_+ I_n \succeq 0$, we obtain that

$$\begin{aligned}
 g_k^\top s_k^{neig} + \frac{1}{2}(s_k^{neig})^\top H_k s_k^{neig} &\leq (r_k^{neig})^\top s_k^{neig} - (s_k^{neig})^\top (H_k + [-\lambda_{\min}(H_k)]_+ I_n) s_k^{neig} \\
 &\quad + \frac{1}{2}(s_k^{neig})^\top (H_k + [-\lambda_{\min}(H_k)]_+ I_n) s_k^{neig} \\
 &\quad - \frac{1}{2}[-\lambda_{\min}(H_k)]_+ \|s_k^{neig}\|^2 - \sqrt{\sigma_k} \|g_k\| \|s_k^{neig}\|^2 \\
 &\leq (r_k^{neig})^\top s_k^{neig} - \frac{1}{2}(s_k^{neig})^\top (H_k + [-\lambda_{\min}(H_k)]_+ I_n) s_k^{neig} \\
 &\quad - \frac{1}{2}[-\lambda_{\min}(H_k)]_+ \|s_k^{neig}\|^2 - \sqrt{\sigma_k} \|g_k\| \|s_k^{neig}\|^2 \\
 &\leq \varsigma_3 \sqrt{\sigma_k} \|g_k\| \|s_k^{neig}\|^2 - \frac{1}{2}[-\lambda_{\min}(H_k)]_+ \|s_k^{neig}\|^2 - \sqrt{\sigma_k} \|g_k\| \|s_k^{neig}\|^2,
 \end{aligned}$$

where we have used (2.10) to obtain the last inequality. Rearranging and ignoring the $\frac{1}{2}[-\lambda_{\min}(H_k)]_+ \|s_k^{neig}\|^2$ term yields (2.18).

Suppose finally that $k \in \mathcal{S}_k^{curv}$. As (2.11) and (2.12) hold, we deduce that

$$g_k^\top s_k^{curv} + \frac{1}{2}(s_k^{curv})^\top H_k s_k^{curv} \leq \frac{1}{2} \|s_k^{curv}\|^2 v_k^\top H_k v_k \leq -\frac{1}{2} \|s_k^{curv}\|^2 \kappa_C \sqrt{\sigma_k} \|g_k\| \leq -\frac{1}{2} \sigma_k \|s_k^{curv}\|^3,$$

yielding (2.19). \square

3 Complexity analysis for the AN2C and AN2E algorithms

We now turn to analyzing the worst-case complexity of the AN2C algorithm, from which that of AN2E will follow. Our analysis is conducted under the following assumptions.

AS.1 The function f is two times continuously differentiable in \mathbb{R}^n .

AS.2 There exists a constant f_{low} such that $f(x) \geq f_{\text{low}}$ for all $x \in \mathbb{R}^n$.

AS.3 The Hessian of f is globally Lipschitz continuous, that is, there exists a non-negative constant L_H such that

$$\|\nabla_x^2 f(x) - \nabla_x^2 f(y)\| \leq L_H \|x - y\| \text{ for all } x, y \in \mathbb{R}^n. \quad (3.1)$$

AS.4 There exists a constant $\kappa_B > 0$ such that

$$\|\nabla_x^2 f(x)\| \leq \kappa_B \text{ for all } x \in \{y \in \mathbb{R}^n \mid f(y) \leq f(x_0)\}.$$

AS.1-AS.3 are standard assumptions when analyzing algorithms that utilize second-order information [9, 3]. AS.4 is also standard when theoretically analyzing second-order methods that combines negative curvature and gradient based directions [38, 15, 30], although we recognize that requiring AS.3 and AS.4 together is slightly unusual. As it turns out, AS.4 is only needed for x being any iterate x_k produced by the algorithm and these iterates all belong to the level associated with the starting point x_0 because the acceptance condition in Step 3 ensures that the sequence $\{f(x_k)\}$ is non-increasing. If this level set is bounded or if the sequence $\{x_k\}$ remains bounded for any other reason, we immediately obtain that

$$\|H_k\| \leq \kappa_B \text{ for all } k \geq 0 \quad (3.2)$$

for some $\kappa_B \geq 0$, and both AS.3 and AS.4 automatically hold.

Having established a lower bound on the decrease ratio in Lemma 2.3, we next proceed to derive an upper bound on the regularization parameter. This is a crucial step when analysing adaptive regularization methods.

Lemma 3.1 Suppose that AS.1 and AS.3 hold. Then, for all $k \geq 0$,

$$\sigma_k \leq \gamma_3 \max \left(\sigma_0, \varsigma_{\max} \frac{L_H}{6(1-\eta_2)} \right), \quad (3.3)$$

where

$$\varsigma_{\max} \stackrel{\text{def}}{=} \max \left(\frac{(1+\kappa\theta)}{(1-\varsigma_3)}, \frac{2(1+\kappa\theta)}{\kappa_a \varsigma_1 (1-2\varsigma_2)}, 2 \right). \quad (3.4)$$

Proof. Let us compute the ratio ρ_k for $k \in S_k^{\text{neig}}$. By using AS.3 and the standard error bound for Lipschitz approximation of the function (see [11, Lemma 2.1]), (2.18) and (2.15), we obtain that

$$\begin{aligned} 1 - \rho_k &= \frac{f(x_k + s_k) - f(x_k) - g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k}{-(g_k^\top s_k + \frac{1}{2} s_k^\top H_k s_k)} \\ &\geq -\frac{L_H \|s_k^{\text{neig}}\|^3}{6(1-\varsigma_3) \sqrt{\sigma_k} \|g_k\| \|s_k^{\text{neig}}\|^2} \\ &\geq -\frac{L_H \|s_k^{\text{neig}}\|}{6(1-\varsigma_3) \sqrt{\sigma_k} \|g_k\|} \\ &\geq -\frac{L_H(1+\kappa\theta)}{6(1-\varsigma_3)\sigma_k}. \end{aligned} \quad (3.5)$$

Hence, if $\sigma_k \geq \frac{L_H(1+\kappa\theta)}{6(1-\varsigma_3)(1-\eta_2)}$, then $\rho_k \geq \eta_2$, which implies that iteration k is successful and $\sigma_{k+1} \leq \sigma_k$ because of (2.8). The mechanism of (2.8) in the algorithm then ensures that

$$\sigma_k \leq \gamma_3 \max \left(\sigma_0, \frac{L_H(1+\kappa\theta)}{6(1-\varsigma_3)(1-\eta_2)} \right). \quad (3.6)$$

Similarly, if $k \in \mathcal{S}_k^{conv}$, we use AS.3, the Lipschitz approximation error bound, (2.17) and (2.5) to deduce that

$$1 - \rho_k \geq -\frac{L_H \|s_k^{conv}\|}{3(1 - 2\varsigma_2) \sqrt{\kappa_a \sigma_k \|g_k\|}} \geq -\frac{L_H(1 + \kappa_\theta)}{3\kappa_a \varsigma_1(1 - 2\varsigma_2)\sigma_k}.$$

Using the same argument as above, we now obtain that

$$\sigma_k \leq \gamma_3 \max \left(\sigma_0, \frac{L_H(1 + \kappa_\theta)}{3\kappa_a \varsigma_1(1 - 2\varsigma_2)(1 - \eta_2)} \right). \quad (3.7)$$

Consider finally the case where $k \in \mathcal{S}_k^{curv}$. Again using AS.3, the Lipschitz approximation error bound and (2.19) lower-bound, we derive that

$$1 - \rho_k = \frac{f(x_k + s_k) - f(x_k) - g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k}{-g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k} \geq \frac{-L_H \|s_k^{curv}\|^3}{6\frac{1}{2}\sigma_k \|s_k^{curv}\|^3} = \frac{-L_H}{3\sigma_k},$$

so that

$$\sigma_k \leq \gamma_3 \max \left(\sigma_0, \frac{L_H}{3(1 - \eta_2)} \right). \quad (3.8)$$

Combining (3.6), (3.7) and (3.8) gives (3.3) with ς_{\max} defined by (3.4). \square

We now prove a lower bound on the decrease at a successful iteration k using negative curvature. We will also bound the change in the norm $\|g_{k+1}\|$ in term of $\|g_k\|$, which will be useful later to bound the cardinal of a subset of $\mathcal{S}_k^{neig} \cup \mathcal{S}_k^{curv}$.

Lemma 3.2 Suppose that AS.1, AS.3 and AS.4 hold and that $k \in \mathcal{S}_k^{curv}$ before termination. Then

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 \kappa_C^3}{2\sqrt{\sigma_{\max}}} \epsilon^{\frac{3}{2}}, \quad (3.9)$$

and

$$\|g_{k+1}\| \leq \left(\frac{L_H}{2\sigma_k} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\epsilon \sigma_k}} + 1 \right) \|g_k\|. \quad (3.10)$$

Proof. Let $k \in \mathcal{S}_k^{curv}$. From (2.7) and (2.19), we obtain that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \left(-g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k \right) \geq \frac{\eta_1}{2} \sigma_k \|s_k^{curv}\|^3.$$

Using now that $\|s_k^{curv}\|^3 = \frac{\kappa_C^3 (\sigma_k \|g_k\|)^{\frac{3}{2}}}{\sigma_k^3}$ (see (2.12)) in the previous inequality gives that

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 \kappa_C^3}{2\sqrt{\sigma_k}} \|g_k\|^{\frac{3}{2}}.$$

Since $\|g_k\| \geq \epsilon$ before termination and that $\sigma_k \leq \sigma_{\max}$ by Lemma 3.1, we obtain (3.9).

Let us now prove (3.10). By using the Lipschitz error bound for the gradient ([11, Lemma 2.1]), the triangular inequality, the fact that $k \in \mathcal{S}_k^{curv}$, (2.11), (2.12), (3.2) (resulting from AS.4) and that before termination $\|g_k\| \geq \epsilon$, we obtain that

$$\begin{aligned}
 \|g_{k+1}\| &\leq \|g_{k+1} - g_k - H_k s_k\| + \|H_k s_k + g_k\| \\
 &\leq \frac{L_H}{2} \|s_k\|^2 + \|g_k\| + \|H_k s_k\| \\
 &\leq \frac{L_H}{2} \|s_k^{curv}\|^2 + \|g_k\| + \|H_k s_k^{curv}\| \\
 &\leq \frac{L_H}{2} \|s_k^{curv}\|^2 + \|g_k\| + \kappa_B \|s_k^{curv}\| \\
 &\leq \frac{L_H}{2\sigma_k} \kappa_C^2 \|g_k\| + \|g_k\| + \kappa_B \kappa_C \frac{\sqrt{\|g_k\|}}{\sqrt{\sigma_k}} \\
 &\leq \left(\frac{L_H}{2\sigma_k} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\sigma_k} \|g_k\|} + 1 \right) \|g_k\| \\
 &\leq \left(\frac{L_H}{2\sigma_k} \kappa_C^2 + \frac{\kappa_C \kappa_B}{\sqrt{\sigma_k} \epsilon} + 1 \right) \|g_k\|,
 \end{aligned}$$

giving (3.10). □

This lemma is the only result requiring AS.4 or its weaker formulation (3.2). Moreover, this assumption is only needed to bound $\|H_k s_k^{curv}\|$ and is therefore required only along directions of negative curvature (which we expect to occur rarely in practice).

After proving a lower bound on the quadratic's decrease when $k \in \mathcal{S}_k^{curv}$, we now exhibit a relationship between the decrease on the objective function decrease and gradient both at iteration k and $k + 1$ for $k \in \mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}$. Moreover, we also prove an inequality between the norms of the gradient at two successive iterations, similar to (3.10).

Lemma 3.3 Suppose that AS.1 and AS.3 hold and that $k \in \mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}$ before termination. Then

$$\|g_{k+1}\| \leq \left(\frac{L_H(1 + \kappa_\theta)}{2\varsigma_1^2 \sigma_k} + \frac{2}{\varsigma_1} + \kappa_C \right) (1 + \kappa_\theta) \|g_k\| \quad (3.11)$$

and

$$\begin{aligned}
 f(x_k) - f(x_{k+1}) &\geq \eta_1 \varsigma_{\min} \sqrt{\sigma_k} \|g_k\| \\
 &\quad \left(\frac{-(2 + \kappa_C) \sqrt{\kappa_a \sigma_k} \|g_k\| + \sqrt{(2 + \kappa_C)^2 \kappa_a \sigma_k \|g_k\| + 2L_H \|g_{k+1}\|}}{L_H} \right)^2
 \end{aligned} \quad (3.12)$$

where

$$\varsigma_{\min} \stackrel{\text{def}}{=} \min \left(\frac{1 - 2\varsigma_2}{2}, 1 - \varsigma_3 \right). \quad (3.13)$$

Proof. Consider first the case where $k \in \mathcal{S}_k^{neig}$. By using the Lipschitz error bound for the gradient, the shape of (2.14), the condition on the residuals (2.10) and the fact that $[-\lambda_{\min}(H_k)]_+ \leq \kappa_C \sqrt{\sigma_k \|g_k\|}$ for $k \in \mathcal{S}_k^{neig}$, we deduce that

$$\begin{aligned} \|g_{k+1}\| &\leq \|g_{k+1} - H_k s_k^{neig} - g_k\| + \|H_k s_k^{neig} + g_k\| \\ &\leq \frac{L_H}{2} \|s_k^{neig}\|^2 + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+) \|s_k^{neig}\| + \|r_k^{neig}\| \\ &\leq \frac{L_H}{2} \|s_k^{neig}\|^2 + (1 + \kappa_C) \sqrt{\sigma_k \|g_k\|} \|s_k^{neig}\| + \varsigma_3 \sqrt{\sigma_k \|g_k\|} \|s_k^{neig}\|. \end{aligned} \quad (3.14)$$

Using now (2.15) yields that

$$\|g_{k+1}\| \leq \left(\frac{L_H}{2\sigma_k} (1 + \kappa_\theta) + 1 + \varsigma_3 + \kappa_C \right) (1 + \kappa_\theta) \|g_k\|. \quad (3.15)$$

Consider now $k \in \mathcal{S}_k^{conv}$. By arguments similar to those used for (3.14), this time with (2.16) and (2.3), we obtain that

$$\begin{aligned} \|g_{k+1}\| &\leq \|g_{k+1} - H_k s_k^{conv} - g_k\| + \|H_k s_k^{conv} + g_k\| \\ &\leq \frac{L_H}{2} \|s_k^{conv}\|^2 + \sqrt{\kappa_a \sigma_k \|g_k\|} \|s_k^{conv}\| + \|r_k^{conv}\| \\ &\leq \frac{L_H}{2} \|s_k^{conv}\|^2 + \sqrt{\kappa_a \sigma_k \|g_k\|} \|s_k^{conv}\| + \varsigma_2 \sqrt{\kappa_a \sigma_k \|g_k\|} \|s_k^{conv}\|. \end{aligned} \quad (3.16)$$

Injecting (2.5) in the last inequality yields that

$$\|g_{k+1}\| \leq \left(\frac{L_H(1 + \kappa_\theta)}{2\varsigma_1^2 \kappa_a \sigma_k} + \frac{1 + \varsigma_2}{\varsigma_1} \right) (1 + \kappa_\theta) \|g_k\|, \quad (3.17)$$

so that taking the larger bound for both (3.15) and (3.17) and using the bounds $\varsigma_2 < \frac{1}{2}$ and $\varsigma_3 < 1$ gives (3.11).

Finally, from (3.16), (3.14), the bounds $\max(\varsigma_3, \varsigma_2) < 1$ and $\kappa_a \geq 1$, we obtain that, for $k \in \mathcal{S}_k^{conv} \cup \mathcal{S}_k^n$,

$$\frac{L_H}{2} \|s_k\|^2 + (2 + \kappa_C) \sqrt{\kappa_a \sigma_k \|g_k\|} \|s_k\| - \|g_{k+1}\| \geq 0.$$

Hence $\|s_k\|$ is larger than the positive root of this quadratic and therefore

$$\|s_k\| \geq \frac{-(2 + \kappa_C) \sqrt{\kappa_a \sigma_k \|g_k\|} + \sqrt{(2 + \kappa_C)^2 \kappa_a \sigma_k \|g_k\| + 2L_H \|g_{k+1}\|}}{L_H} > 0.$$

We then deduce (3.12) from this inequality, (2.7), the lower bounds on the quadratic decrease for $k \in \mathcal{S}_k^{neig}$ or $k \in \mathcal{S}_k^{conv}$ ((2.18) and (2.17) respectively) and the definition of ς_{\min} in (3.13). \square

The bound (3.12) is not sufficient for deriving the required $\mathcal{O}(\epsilon^{-3/2})$ optimal complexity rate because the decrease depends on both $\|g_{k+1}\|$ and $\|g_k\|$. Indeed, when $\|g_{k+1}\| \ll \|g_k\|$, the left-hand side of (3.12) tends to zero. To circumvent this difficulty, the next lemma borrows some elements of [33, Theorem 1] and partitions $\mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}$ in two further subsets. The

minimum decrease on the objective function is of the required magnitude in the first one while no meaningful information can be derived on the decrease on the function value in the second, albeit the magnitude of the gradient at the next iteration is halved. The bounds (3.11) and (3.10) are then used to bound the cardinal of the latter set.

Lemma 3.4 Suppose that AS.1, AS.3 and AS.4 hold and that $\mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}$ is partitioned as

$$\mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv} = \mathcal{S}_k^{decr} \cup \mathcal{S}_k^{divgrad} \quad (3.18)$$

where

$$\mathcal{S}_k^{decr} \stackrel{\text{def}}{=} \{k \in \mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}, \sigma_k \|g_k\| \leq \kappa_m 2L_H \|g_{k+1}\|\}, \quad (3.19)$$

$$\mathcal{S}_k^{divgrad} \stackrel{\text{def}}{=} \{k \in \mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}, \sigma_k \|g_k\| > \kappa_m 2L_H \|g_{k+1}\|\} \quad (3.20)$$

with

$$\kappa_m \stackrel{\text{def}}{=} \gamma_3 \max \left(\frac{\sigma_0}{L_H}, \frac{\varsigma_{\max}}{6(1-\eta_2)} \right). \quad (3.21)$$

Then, for all $k \in \mathcal{S}_k^{decr}$,

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 \varsigma_{\min}(\sigma_k \|g_k\|)^{\frac{3}{2}}}{\left(\kappa_m L_H \left((2 + \kappa_C) \sqrt{\kappa_a} + \sqrt{(2 + \kappa_C)^2 \kappa_a + \frac{1}{\kappa_m}} \right) \right)^2}. \quad (3.22)$$

Moreover,

$$|\mathcal{S}_k^{divgrad}| \leq \kappa_n |\mathcal{S}_k^{decr}| + \left(\frac{1}{2 \log(2)} |\log(\epsilon)| + \kappa_{curv} \right) |\mathcal{S}_k^{curv}| + \frac{|\log(\epsilon)| + \log(\|g_0\|)}{\log(2)} + 1, \quad (3.23)$$

where

$$\kappa_n \stackrel{\text{def}}{=} \frac{\log \left(\frac{L_H(1+\kappa_\theta)}{2\varsigma_1^2 \sigma_{\min}} + \frac{2}{\varsigma_1} + \kappa_C \right) + \log(1 + \kappa_\theta)}{\log(2)}, \quad \kappa_{curv} \stackrel{\text{def}}{=} \frac{\log \left(\frac{L_H}{2\sigma_{\min}} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\sigma_{\min}}} + 1 \right)}{\log(2)}. \quad (3.24)$$

Proof. Let $k \in \mathcal{S}_k^{decr}$. Injecting the definition of \mathcal{S}_k^{decr} (3.19) in (3.12), we obtain that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \varsigma_{\min}(\sigma_k \|g_k\|)^{\frac{3}{2}} \left(\frac{-(2 + \kappa_C) \sqrt{\kappa_a} + \sqrt{\kappa_a (2 + \kappa_C)^2 + \frac{1}{\kappa_m}}}{L_H} \right)^2.$$

Taking the conjugate both at the denominator and numerator yields (3.22).

Let $k \in \mathcal{S}_k^{divgrad}$. Using the definition of κ_m in (3.21) and that of $\mathcal{S}_k^{divgrad}$ in (3.20) gives that

$$\|g_{k+1}\| < \frac{\sigma_k}{\kappa_m L_H} \frac{\|g_k\|}{2} \leq \frac{\sigma_k}{\gamma_3 \max \left(\frac{\sigma_0}{L_H}, \frac{\varsigma_{\max}}{6(1-\eta_2)} \right) L_H} \frac{\|g_k\|}{2} \leq \frac{\|g_k\|}{2}, \quad (3.25)$$

where the last inequality results from the upper bound on σ_k in (3.3).

Successively using the fact that $\mathcal{S}_k = \mathcal{S}_k^{decr} \cup \mathcal{S}_k^{divgrad} \cup \mathcal{S}_k^{curv}$, the relationship between $\|g_{k+1}\|$ and $\|g_k\|$ in the three cases ((3.25), (3.11) and (3.10)), the fact that $\sigma_k \geq \sigma_{\min}$ in (3.11) and (3.10), we then deduce that

$$\begin{aligned} \frac{\epsilon}{\|g_0\|} &\leq \frac{\|g_k\|}{\|g_0\|} = \prod_{i \in \mathcal{S}_k \setminus \{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \\ &= \prod_{i \in \mathcal{S}_k^{decr} \setminus \{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \prod_{i \in \mathcal{S}_k^{divgrad} \setminus \{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \prod_{i \in \mathcal{S}_k^{curv} \setminus \{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \\ &\leq \left[\left(\frac{L_H(1 + \kappa_\theta)}{2\varsigma_1^2 \sigma_{\min}} + \frac{2}{\varsigma_1} + \kappa_C \right) (1 + \kappa_\theta) \right]^{|\mathcal{S}_k^{decr} \setminus \{k\}|} \times \frac{1}{2^{|\mathcal{S}_k^{divgrad} \setminus \{k\}|}} \times \\ &\quad \left[\frac{L_H}{2\sigma_{\min}} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\epsilon \sigma_{\min}}} + 1 \right]^{|\mathcal{S}_k^{curv} \setminus \{k\}|}. \end{aligned}$$

Now $\varsigma_1 \leq 1$ and thus both terms in brackets are larger than one. Moreover, obviously, $|\mathcal{S}_k^{decr} \setminus \{k\}| \leq |\mathcal{S}_k^{decr}|$ and $|\mathcal{S}_k^{curv} \setminus \{k\}| \leq |\mathcal{S}_k^{curv}|$, so that

$$\frac{2^{|\mathcal{S}_k^{divgrad} \setminus \{k\}|} \epsilon}{\|g_0\|} \leq \left[\left(\frac{L_H(1 + \kappa_\theta)}{2\varsigma_1^2 \sigma_{\min}} + \frac{2}{\varsigma_1} + \kappa_C \right) (1 + \kappa_\theta) \right]^{|\mathcal{S}_k^{decr}|} \left[\frac{L_H}{2\sigma_{\min}} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\epsilon \sigma_{\min}}} + 1 \right]^{|\mathcal{S}_k^{curv}|}.$$

Taking logarithms gives that

$$\begin{aligned} |\mathcal{S}_k^{divgrad} \setminus \{k\}| \log(2) &\leq \log \left[\left(\frac{L_H(1 + \kappa_\theta)}{2\varsigma_1^2 \sigma_{\min}} + \frac{2}{\varsigma_1} + \kappa_C \right) (1 + \kappa_\theta) \right] |\mathcal{S}_k^{decr}| + \log(\|g_0\|) \\ &\quad + |\log(\epsilon)| + \log \left[\frac{L_H}{2\sigma_{\min}} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\epsilon \sigma_{\min}}} + 1 \right] |\mathcal{S}_k^{curv}|. \end{aligned}$$

We then obtain (3.23) with the values of κ_n and κ_{curv} stated in (3.24) by dividing this last inequality by $\log(2)$ and using the facts that $|\mathcal{S}_k^{divgrad} \setminus \{k\}| \geq |\mathcal{S}_k^{divgrad}| - 1$ and $\frac{1}{\sqrt{\epsilon}} \geq 1$. \square

Combining the previous lemmas, we are now able to state the complexity of the AN2C algorithm. Our theorem statement relies on the observation that the objective function is evaluated once per iteration, and its derivatives once per successful iteration.

Theorem 3.5 Suppose that AS.1- AS.4 hold. Then the AN2C algorithm requires at most

$$|\mathcal{S}_k| \leq \kappa_\star |\log(\epsilon)| \epsilon^{-\frac{3}{2}}$$

successful iterations and evaluations of the gradient and the Hessian and at most

$$\kappa_\star \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) |\log(\epsilon)| \epsilon^{-\frac{3}{2}} + \frac{1}{\log \gamma_3} \log \left(\frac{\sigma_{\max}}{\sigma_0} \right)$$

evaluations of f to produce a vector x_ϵ such that $\|g(x_\epsilon)\| \leq \epsilon$ where κ_\star is a constant only depending on the problem.

Proof. First note that we only need to prove an upper bound on $|\mathcal{S}_k^{decr}|$ and $|\mathcal{S}_k^{curv}|$ to derive a bound on $|\mathcal{S}_k|$ since

$$|\mathcal{S}_k| = |\mathcal{S}_k^{decr}| + |\mathcal{S}_k^{curv}| + |\mathcal{S}_k^{divgrad}| \quad (3.26)$$

and a bound on $|\mathcal{S}_k^{divgrad}|$ is given by (3.23). We start by proving an upper bound on $|\mathcal{S}_k^{curv}|$. Using AS.2 and the lower bound on the decrease of the function values (3.9), we derive that, for $k \in \mathcal{S}_k^{curv}$,

$$f(x_0) - f_{\text{low}} = \sum_{i \in \mathcal{S}_k} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{curv}} f(x_i) - f(x_{i+1}) \geq |\mathcal{S}_k^{curv}| \frac{\eta_1 \kappa_C^3}{2\sqrt{\sigma_{\max}}} \epsilon^{\frac{3}{2}}$$

and hence that

$$|\mathcal{S}_k^{curv}| \leq \frac{2(f(x_0) - f_{\text{low}})\sqrt{\sigma_{\max}}}{\eta_1 \kappa_C^3} \epsilon^{-\frac{3}{2}}. \quad (3.27)$$

Similarly for $k \in \mathcal{S}_k^{decr}$, using AS.2, (3.22), the fact that $\sigma_k \geq \sigma_{\min}$ and $\|g_k\| \geq \epsilon$ before termination yields that

$$f(x_0) - f_{\text{low}} \geq \sum_{i \in \mathcal{S}_k^{decr}} f(x_i) - f(x_{i+1}) \geq \frac{|\mathcal{S}_k^{decr}| \eta_1 \sigma_{\min} (\sigma_{\min} \epsilon)^{\frac{3}{2}}}{\left(L_H \kappa_m (\sqrt{\kappa_a} (2 + \kappa_C) + \sqrt{\kappa_a (2 + \kappa_C)^2 + \frac{1}{\kappa_m}})\right)^2}$$

where κ_m is defined in (3.21). Rearranging the last inequality yields that

$$|\mathcal{S}_k^{decr}| \leq \frac{\left(L_H \kappa_m (\sqrt{\kappa_a} (2 + \kappa_C) + \sqrt{\kappa_a (2 + \kappa_C)^2 + \frac{1}{\kappa_m}})\right)^2}{\eta_1 \sigma_{\min} \epsilon^{\frac{3}{2}}} \epsilon^{-\frac{3}{2}}. \quad (3.28)$$

Combining now (3.27) and (3.28) with the upper-bound (3.23) on $|\mathcal{S}_k^{divgrad}|$, we deduce that

$$|\mathcal{S}_k^{divgrad}| \leq \kappa_{divgrad} |\log(\epsilon)| \epsilon^{-\frac{3}{2}} \quad (3.29)$$

where $\kappa_{divgrad}$ only depends on problem constants. Summing (3.27), (3.28), (3.29) to bound $|\mathcal{S}_k|$ in (3.26) then gives that

$$|\mathcal{S}_k| \leq \kappa_* |\log(\epsilon)| \epsilon^{-\frac{3}{2}}, \quad (3.30)$$

proving the first part of the theorem. The second part is then deduced from (3.30) combined with Lemma 2.1. \square

The $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3/2})$ complexity order in ϵ only differs by the factor $|\log(\epsilon)|$ from the optimal order for nonconvex second-order methods [10], a factor which is typically small for practical values of ϵ . The AN2C algorithm thus enjoys a better complexity order than that of past hybrid algorithms [15, 30, 21] for which the order is $\mathcal{O}(\epsilon^{-2})$. However, it is marginally worse than that of the more complex second-order linesearch of [38] which attains the optimal order. Moreover, we see in the proof of Theorem 3.5 that the $|\log \epsilon|$ term appears because of (3.23) and (3.27) and we may hope that the number of s_k^{curv} iterations is typically much less than its worst-case $\mathcal{O}(\epsilon^{-3/2})$ in practice. The trust-region algorithm of [16] has the same complexity as AN2C.

We conclude this analysis by noting that Theorem 3.5 also applies to the AN2E algorithm introduced on page 6. Indeed trial steps are then computed solely via (2.6) in this method and the fact that \mathcal{S}^{conv} is therefore empty does not affect our arguments. Hence the $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3/2})$ complexity order in ϵ also holds for the AN2E algorithm.

4 Finding second-order critical points

Can the AN2C algorithm be strengthened to ensure it will compute second-order critical points? We show in this section under the same assumptions as that used for its first-order analysis. The resulting modified algorithm, which we call SOAN2C (for Second-Order AN2C) makes extensive use of the Step of AN2C, and is detailed on the current page.

Algorithm 4.1: Second Order Adaptive Newton with Negative Curvature (SOAN2C)

Step 0: Initialization: Identical to AN2C[Step 0] with $\epsilon \in (0, 1]$ now replaced by $\epsilon_1 \in (0, 1]$ and $\epsilon_2 \in (0, 1]$.

Step 1: Compute current derivatives Evaluate g_k and H_k . Terminate if

$$\|g_k\| \leq \epsilon_1 \text{ and } \lambda_{\min}(H_k) \geq -\epsilon_2. \quad (4.1)$$

Step 2: Step calculation If $\|g_k\| > \epsilon_1$,

$$s_k = s_k^{fo} \stackrel{\text{def}}{=} \text{AN2C}[\text{Step2}](g_k, H_k, \sigma_k, \kappa_a, \kappa_C, \kappa_\theta, \varsigma_1, \varsigma_2, \varsigma_3). \quad (4.2)$$

Else compute v_k such that

$$g_k^\top v_k \leq 0, \quad \|v_k\| = 1 \text{ and } H v_k = \lambda_{\min}(H_k) v_k, \quad (4.3)$$

and set

$$s_k = s_k^{so} \stackrel{\text{def}}{=} \frac{-\lambda_{\min}(H_k)}{\sigma_k} v_k. \quad (4.4)$$

Step 4: Acceptance ratio computation Identical to AN2C[Step 4].

Step 5: Regularization parameter update Identical to AN2C[Step 5].

Because the step may be computed using (4.2), the notations defining the partitions of $|\mathcal{S}_k|$ are still relevant, but we complete them by introducing

$$\mathcal{S}^{so} \stackrel{\text{def}}{=} \mathcal{S} \cap \{s_k = s_k^{so}\}, \quad \mathcal{S}_k^{so} \stackrel{\text{def}}{=} \mathcal{S}_k \cap \{s_k = s_k^{so}\}, \quad \mathcal{S}^{fo} \stackrel{\text{def}}{=} \mathcal{S} \setminus \mathcal{S}^{so} \text{ and } \mathcal{S}_k^{fo} \stackrel{\text{def}}{=} \mathcal{S}_k \setminus \mathcal{S}_k^{so}.$$

In addition, for $m \geq n \geq 0$, we define

$$\mathcal{S}_{n,m} \stackrel{\text{def}}{=} \mathcal{S} \cap \{n, \dots, m\}$$

and we naturally extend this notation using superscripts identifying the subsets of $\mathcal{S}_{n,m}$ corresponding to the different iteration types identified above. We also introduce two index sequences whose purpose is to keep track of when $s_k = s_k^{fo}$ (4.2) or $s_k = s_k^{so}$ (4.4), in the sense that

$$s_k = s_k^{fo} \text{ for } k \in \bigcup_{i \geq 0, p_i \geq 0} \{p_i, \dots, q_i - 1\} \text{ and } s_k = s_k^{so} \text{ for } k \in \bigcup_{i \geq 0} \{q_i, \dots, p_{i+1} - 1\}.$$

Formally,

$$p_0 = \begin{cases} 0 & \text{if } \|g_0\| > \epsilon_1 \\ -1 & \text{if } \|g_0\| \leq \epsilon_1, \end{cases} \quad \text{and} \quad q_0 = \begin{cases} \inf\{k > 0 \mid \|g_k\| \leq \epsilon_1\} & \text{if } \|g_0\| > \epsilon_1 \\ 0 & \text{if } \|g_0\| \leq \epsilon_1. \end{cases} \quad (4.5)$$

Then

$$p_i \stackrel{\text{def}}{=} \inf\{k > q_{i-1} \mid \|g_k\| > \epsilon_1\} \quad \text{and} \quad q_i \stackrel{\text{def}}{=} \inf\{k > p_i \mid \|g_k\| \leq \epsilon_1\} \quad \text{for } i \geq 1. \quad (4.6)$$

The following lemma states an important decrease property holding when (4.4) is used. We also verify that the bound on the regularization parameter derived in the last section still applies.

Lemma 4.1 Suppose that AS.1 and AS.3 hold. Let $k \in \mathcal{S}^{so}$. Then

$$-g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k \geq \frac{1}{2} \sigma_k \|s_k\|^3. \quad (4.7)$$

Moreover, the upper bound (3.3) still holds for all $k \geq 0$.

Proof. We obtain from (4.3) and (4.4) that

$$g_k^\top s_k^{so} + \frac{1}{2} (s_k^{so})^\top H_k s_k^{so} \leq \frac{1}{2} \|s_k^{so}\|^2 v_k^\top H_k v_k = \frac{1}{2} \|s_k^{so}\|^2 \lambda_{\min}(H_k) \leq -\frac{1}{2} \sigma_k \|s_k^{so}\|^3,$$

which gives (4.7). As in Lemma 3.1, we now use AS.3, the standard Lipschitz error bound for the function (see [11, Lemma 2.1]) and (4.7) to deduce that

$$1 - \rho_k = \frac{f(x_k + s_k) - f(x_k) - g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k}{-g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k} \geq \frac{-L_H \|s_k^{so}\|^3}{6(\frac{1}{2} \sigma_k \|s_k^{so}\|^3)} = \frac{-L_H}{3\sigma_k},$$

Thus, if $\sigma_k \geq \frac{L_H}{3(1-\eta_2)}$, we have that $\rho_k \geq \eta_2$ and k is a successful iteration. We may then use the argument of Lemma 3.1 and the fact that ς_{\max} introduced in (3.4) is larger than two to deduce that (3.3) also holds for the SOAN2C algorithm. \square

We now prove a lemma analogue to (3.2) but this time using a negative-curvature step as described in (4.3)-(4.4). We will also bound the sequence of $\|g_{p_i}\|$.

Lemma 4.2 Suppose that AS.1, AS.3 and AS.4 hold. Then, for $k \in \mathcal{S}^{so}$,

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{2\sigma_{\max}^2} \epsilon_2^3. \quad (4.8)$$

We also have that

$$\|g_{p_i}\| \leq \kappa_{gpi} \stackrel{\text{def}}{=} \max \left[\|g_0\|, \left(\frac{L_H \kappa_B^2}{2\sigma_{p_i-1}^2} + \frac{\kappa_B^2}{\sigma_{p_i-1}} + 1 \right) \right]. \quad (4.9)$$

for all $p_i \geq 0$ as defined in (4.5)-(4.6).

Proof. Let $k \in \mathcal{S}^{so}$. From (2.7) and (4.7), we obtain that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \left(-g_k^\top s_k - \frac{1}{2} s_k^\top H_k s_k \right) \geq \frac{\eta_1}{2} \sigma_k \|s_k^{so}\|^3.$$

Using now that $\|s_k^{so}\|^3 = \frac{|\lambda_{\min}(H_k)|^3}{\sigma_k^3}$ (see (4.4)) in the previous inequality gives that

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{2\sigma_k^2} |\lambda_{\min}(H_k)|^3.$$

Now $|\lambda_{\min}(H_k)| \geq \epsilon_2$ when s_k^{so} is computed and $\sigma_k \leq \sigma_{\max}$ by Lemma 4.1, from which (4.8) follows. Observe now that (4.9) trivially holds if $p_i = p_0 = 0$. Consider now $p_i > 0$. From the definition of p_i and q_i in (4.6), we see that $p_i - 1 \in \mathcal{S}^{so}$. Using the Lipschitz error bound for the gradient ([11, Lemma 2.1]), the triangular inequality (4.3), (4.4), (3.2) (resulting from AS.4), we obtain that

$$\begin{aligned} \|g_{p_i}\| &\leq \|g_{p_i} - g_{p_i-1} - H_{p_i-1} s_{p_i-1}\| + \|H_{p_i-1} s_{p_i-1}^{so} + g_{p_i-1}\| \\ &\leq \frac{L_H}{2} \|s_{p_i-1}^{so}\|^2 + \|g_{p_i-1}\| + \|H_{p_i-1} s_{p_i-1}^{so}\| \\ &\leq \frac{L_H |\lambda_{\min}(H_{p_i-1})|^2}{2\sigma_{p_i-1}^2} + \|g_{p_i-1}\| + \frac{\kappa_B |\lambda_{\min}(H_{p_i-1})|}{\sigma_{p_i-1}} \\ &\leq \frac{L_H \kappa_B^2}{2\sigma_{p_i-1}^2} + \|g_{p_i-1}\| + \frac{\kappa_B^2}{\sigma_{p_i-1}}. \end{aligned}$$

But $\|g_{p_i-1}\| \leq \epsilon_1 \leq 1$ since $p_i - 1 \in \mathcal{S}^{so}$, which gives (4.9). \square

In addition to this lemma, all properties of the different steps derived in Section 3 remain valid because these steps are only computed for $\|g_k\| > \epsilon_1$. In particular, (3.10) still applies with $\epsilon = \epsilon_1$. However, (3.23) in Lemma 3.4 may no longer hold because its proof relies on the fact that $\|g_k\| \geq \epsilon_1$, which is no longer true. The purpose of the next lemma is to provide an analogue of (3.23) for the case where SOAN2C is used by revisiting and completing its proof.

Lemma 4.3 Suppose that AS.1, AS.3 and AS.4 hold and the SOAN2C algorithm is used. Consider the partition of $\mathcal{S}_k^{neig} \cup \mathcal{S}_k^{conv}$ into $\mathcal{S}_k^{decr} \cup \mathcal{S}_k^{divgrad}$ defined in Lemma 3.4 with the same κ_m (defined in (3.21)). Then (3.22) holds for all $k \in \mathcal{S}_k^{decr}$. Moreover,

$$\begin{aligned} |\mathcal{S}_k^{divgrad}| &\leq \kappa_n |\mathcal{S}_k^{decr}| + \left(\frac{1}{2 \log(2)} |\log(\epsilon_1)| + \kappa_{curv} \right) |\mathcal{S}_k^{curv}| \\ &\quad + \left(\frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1 \right) (|\mathcal{S}_k^{so}| + 1) \end{aligned} \quad (4.10)$$

where κ_n and κ_{curv} are defined in (3.24) and κ_{gpi} is given by (4.9).

Proof. The proof of (3.22) is identical to that used in Lemma 3.4. Moreover, we still obtain (3.25) for $k \in \mathcal{S}_k^{divgrad}$, because the definition of κ_m in (3.21) is unchanged and Lemma 4.1 ensures that (3.3) continues to hold for the SOAN2C algorithm.

We now prove (4.10). If \mathcal{S}_k^{fo} is empty, then so is its subset $\mathcal{S}_k^{divgrad}$ and (4.10) trivially holds. If \mathcal{S}_k^{fo} is not empty, we see from the definitions (4.5)-(4.6) that, for some $m \geq 0$ depending on k ,

$$\mathcal{S}_k^{fo} = \{0, \dots, k\} \cap \{\|g_k\| > \epsilon_1\} = \left(\bigcup_{i=0, p_i \geq 0}^{m-1} \{p_i, \dots, q_i - 1\} \right) \cup \{p_m, \dots, k\}. \quad (4.11)$$

Note that the last set in this union is empty unless $k \in \mathcal{S}^{fo}$, in which case $p_m \geq 0$. Suppose first that the set of indices corresponding to the union in brackets is non-empty and let i be an index in this set. Moreover, suppose also that $p_i < q_i - 1$. Using (4.9) and the facts that $\|g_{q_i-1}\| > \epsilon_1$, that the gradient only changes at successful iterations and that $\mathcal{S}_{p_i, q_i-2} = \mathcal{S}_{p_i, q_i-2}^{curv} \cup \mathcal{S}_{p_i, q_i-2}^{divgrad} \cup \mathcal{S}_{p_i, q_i-2}^{decr}$, we now derive that

$$\begin{aligned} \frac{\epsilon_1}{\kappa_{gpi}} &\leq \frac{\|g_{q_i-1}\|}{\|g_{p_i}\|} = \prod_{j=p_i}^{q_i-2} \frac{\|g_{j+1}\|}{\|g_j\|} = \prod_{j \in \mathcal{S}_{p_i, q_i-2}} \frac{\|g_{j+1}\|}{\|g_j\|} \\ &= \prod_{j \in \mathcal{S}_{p_i, q_i-2}^{decr}} \frac{\|g_{j+1}\|}{\|g_j\|} \prod_{j \in \mathcal{S}_{p_i, q_i-2}^{curv}} \frac{\|g_{j+1}\|}{\|g_j\|} \prod_{j \in \mathcal{S}_{p_i, q_i-2}^{divgrad}} \frac{\|g_{j+1}\|}{\|g_j\|} \\ &\leq \left(\left(\frac{L_H(1 + \kappa_\theta)}{2\varsigma_1^2 \sigma_{\min}} + \frac{2}{\varsigma_1} + \kappa_C \right) (1 + \kappa_\theta) \right)^{|\mathcal{S}_{p_i, q_i-2}^{decr}|} \times \frac{1}{2^{|\mathcal{S}_{p_i, q_i-2}^{divgrad}|}} \times \\ &\quad \left(\frac{L_H}{2\sigma_{\min}} \kappa_C^2 + \frac{\kappa_B \kappa_C}{\sqrt{\epsilon_1 \sigma_{\min}}} + 1 \right)^{|\mathcal{S}_{p_i, q_i-2}^{curv}|} \end{aligned}$$

where we used (3.11), (3.10) and (3.25) and the fact that $\sigma_k \geq \sigma_{\min}$ in the last inequality. Rearranging the last inequality, taking the log, using the inequality $|\mathcal{S}_{p_i, q_i-2}^{divgrad}| \geq |\mathcal{S}_{p_i, q_i-1}^{divgrad}| - 1$ and dividing by $\log(2)$ then gives that

$$\left(|\mathcal{S}_{p_i, q_i-1}^{divgrad}| - 1 \right) + \frac{\log(\epsilon_1) - \log(\kappa_{gpi})}{\log(2)} \leq \kappa_n |\mathcal{S}_{p_i, q_i-2}^{decr}| + \left(\frac{|\log(\epsilon_1)|}{2 \log(2)} + \kappa_{curv} \right) |\mathcal{S}_{p_i, q_i-2}^{curv}|$$

with κ_n and κ_{curv} given by (3.24). Further rearranging this inequality and using the fact that $|\mathcal{S}_{p_i, q_i-2}| \leq |\mathcal{S}_{p_i, q_i-1}|$ for the different types of step, we obtain that

$$|\mathcal{S}_{p_i, q_i-1}^{divgrad}| \leq \kappa_n |\mathcal{S}_{p_i, q_i-1}^{decr}| + \left(\frac{|\log(\epsilon_1)|}{2 \log(2)} + \kappa_{curv} \right) |\mathcal{S}_{p_i, q_i-1}^{curv}| + \frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1. \quad (4.12)$$

If now $p_i = q_i - 1$, then clearly $|\mathcal{S}_{p_i, q_i-1}^{divgrad}| \leq 1$ and (4.12) also holds. Using the same reasoning, we derive that, when $\{p_m, \dots, k\}$ is non-empty,

$$|\mathcal{S}_{p_m, k}^{divgrad}| \leq \kappa_n |\mathcal{S}_{p_m, k}^{decr}| + \left(\frac{|\log(\epsilon_1)|}{2 \log(2)} + \kappa_{curv} \right) |\mathcal{S}_{p_m, k}^{curv}| + \frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1, \quad (4.13)$$

and this inequality also holds if $\{p_m, \dots, k\} = \emptyset$ since $\mathcal{S}_{p_m, k}^{divgrad} \subseteq \{p_m, \dots, k\}$. Now adding (4.12) for $i \in \{0, \dots, m\}$ and (4.13) to take (4.11) into account gives that

$$|\mathcal{S}_k^{divgrad}| \leq \kappa_n |\mathcal{S}_k^{decr}| + \left(\frac{|\log(\epsilon_1)|}{2 \log(2)} + \kappa_{curv} \right) |\mathcal{S}_k^{curv}| + \left(\frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1 \right) (m + 1).$$

Observe now each time a q_i is generated, the next successful step must be a second-order step, so that $m \leq |\mathcal{S}_k^{so}|$ and (4.10) follows. \square

Equipped with this last lemma and the results of Sections 3 and 4, we may finally establish the worst-case iteration/evaluation complexity of the SOAN2C algorithm.

Theorem 4.4 Suppose that AS.1–AS.4 hold. Then the SOAN2C algorithm requires at most

$$|\mathcal{S}_k| \leq \kappa_\star |\log(\epsilon_1)| \max(\epsilon_1^{-\frac{3}{2}}, \epsilon_2^{-3})$$

successful iterations and evaluations of the gradient and the Hessian and at most

$$\kappa_\star \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) |\log(\epsilon_1)| \max(\epsilon_1^{-\frac{3}{2}}, \epsilon_2^{-3}) + \frac{1}{\log \gamma_3} \log \left(\frac{\sigma_{\max}}{\sigma_0} \right)$$

evaluations of f to produce a vector x_ϵ such that $\|g(x_\epsilon)\| \leq \epsilon_1$ and $\lambda_{\min}(H_{x_\epsilon}) \geq -\epsilon_2$, where κ_\star is a constant only depending on the problem and σ_{\max} is given by (3.3).

Proof. Note that the bounds (3.27) and (3.28) derived in the proof of Theorem 3.5 are still valid because they only cover steps computed using AN2C, so that we now need to focus on bounding \mathcal{S}_k^{so} . Using AS.2 and the lower bound on the decrease of the function values (3.9), we derive that, for $k \in \mathcal{S}^{so}$,

$$f(x_0) - f_{\text{low}} = \sum_{i \in \mathcal{S}_k} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{so}} f(x_i) - f(x_{i+1}) \geq |\mathcal{S}_k^{so}| \frac{\eta_1}{2\sigma_{\max}^2} \epsilon_2^3,$$

and therefore that

$$|\mathcal{S}_k^{so}| \leq \frac{2\sigma_{\max}^2 (f(x_0) - f_{\text{low}})}{\eta_1} \epsilon_2^{-3}. \quad (4.14)$$

Injecting now (4.14), (3.28) and (3.27) in the bound (4.10) on $\mathcal{S}_k^{\text{divgrad}}$ yields that

$$|\mathcal{S}_k^{\text{divgrad}}| \leq \kappa_1 |\log(\epsilon_1)| \max(\epsilon_1^{-\frac{3}{2}}, \epsilon_2^{-3})$$

where κ_1 is a problem dependent constant. Combining the last inequality with (4.14), (3.28) and (3.27) in $|\mathcal{S}_k| = |\mathcal{S}_k^{\text{divgrad}}| + |\mathcal{S}_k^{\text{curv}}| + |\mathcal{S}_k^{so}| + |\mathcal{S}_k^{\text{decr}}|$ gives that

$$|\mathcal{S}_k| \leq \kappa_\star |\log(\epsilon_1)| \max(\epsilon_1^{-\frac{3}{2}}, \epsilon_2^{-3})$$

proving the first part of the theorem. The second part follows from (3.30) and Lemma 2.1. \square

As for Theorem 3.5, the bound, in which the ϵ_2^{-3} term is likely to dominate, differs from standard one for second-order algorithms seeking second-order points (in $\mathcal{O}(\max(\epsilon_1^{-3/2}, \epsilon_2^{-3}))$) [12, Theorems 3.3.9, 3.4.6] by a (modest) factor $|\log(\epsilon_1)|$. This factor occurs as a consequence of (4.10), (4.14) and (3.27) and one expects that, in practice, (4.14) is smaller than $\mathcal{O}(\epsilon_2^{-3})$ so that Newton steps are taken most often.

As for the first-order case, we note that an SOAN2E algorithm may easily be defined by always using (2.5) for first-order steps in SOAN2C, and that its complexity is covered by Theorem 4.4.

5 Numerical illustration

We now illustrate the behaviour of our proposed algorithms on three sets of test problems from the freely available OPM collection [27]. The first set contains 119 small-dimensional problems, the second contains 75 medium-size ones, while the third contains 59 “largish” ones. The list of problems and their dimensions are listed in Tables 2, 3 and 4 in appendix.

We use Matlab implementations of AN2C and AN2E where the involved linear systems are solved by using the Matlab sparse Cholesky factorization, and where we have set

$$\begin{aligned} \kappa_C = 10^8, \kappa_a = 100, \kappa_\theta = 1, \varsigma_1 = \frac{1}{2}, \varsigma_2 = \varsigma_3 = 10^{-10}, \\ \gamma_1 = \frac{1}{2}, \gamma_2 = \gamma_3 = 10, \eta_1 = 10^{-4} \text{ and } \eta_2 = 0.95. \end{aligned}$$

We compare AN2C and AN2E with implementations of the standard adaptive regularization AR2 and trust-region TR2M, two well-regarded methods. All these algorithms use quadratic approximations of the objective function (i.e. gradients and Hessians). The first three also use the same acceptance thresholds η_1 and η_2 and values of γ_1 , γ_2 and γ_3 . The TR2M method shrinks the trust-region radius by a factor $\sqrt{10}$ and expands it by a factor 2 (see [12, Section 11.2] for a discussion of the coherence of these factors between trust-region and adaptive regularization methods). The authors are aware that further method-dependent tuning would possibly result in improved performance, but the values chosen here appear to work reasonably well for each method. The step computation is performed in AR2 using an (unpreconditioned) Lanczos approach while a standard Moré-Sorensen method⁽¹⁾ is used in TR2M (see [12, Chapter 9] for details). For AR2, the step computation is terminated as soon as

$$\|g_k + H_k s_k\| \leq \frac{1}{2} \theta_{sub} \sigma_k \|s_k\|^2 \quad (5.1)$$

(see [28] for a justification), while the Moré-Sorensen iterations in TR2M are terminated as soon as $\|s_k\| \in [(1 - \theta_{sub})\Delta_k, (1 + \theta_{sub})\Delta_k]$, where, in both cases, $\theta_{sub} = 10^{-3}$ for $n \leq 100$ and 10^{-2} for $n > 100$. All experiments were run on a Dell Precision computer with Matlab 2022b.

We discuss our experiments from the efficiency and reliability points of view. Efficiency is measured, in accordance with the complexity theory, in number of iterations (or, equivalently, function and possibly derivatives’ evaluations): the fewer the more efficient the algorithm. In addition to presenting the now standard performance profiles [18] for our four algorithms in Figure 1, we follow [37, 25] and consider the derived “global” measure π_{algo} to be $\frac{1}{10}$ of the area below the curve corresponding to `algo` in the performance profile, for abscissas in the interval $[1, 10]$. The larger this area and the closer π_{algo} to one, the closer the curve to the right and top borders of the plot and the better the global performance.

When reporting reliability, we say that the run of an algorithmic variant on a specific test problem is successful if the gradient norm tolerance $\epsilon = 10^{-6}$ has been achieved in the allotted cpu-time (1h) and before the maximum number of iterations (5000) is reached. The ρ_{algo} statistic denotes the percentage of successful runs taken on all problems.

⁽¹⁾Given that our version of AN2C uses matrix factorizations, it seems more natural to compare it with a Moré-Sorensen-based trust-region than to one using truncated conjugate gradients.

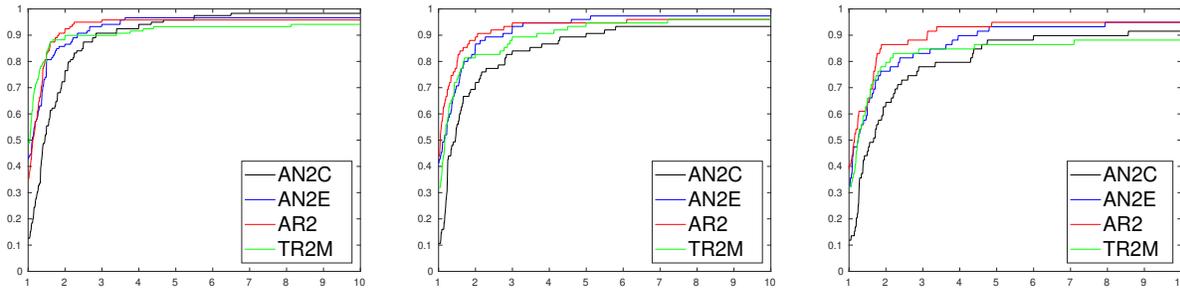


Figure 1: Iteration performance profiles for OPM problems (left: small, center: medium, right: largish). We report on the vertical axis the proportion of problems for which the number of iterations of each algorithm is at most a fraction (given by the horizontal axis) of the smallest across all algorithms (see [18]).

	small pbs.		medium pbs.		largish pbs.	
algo	π_{algo}	ρ_{algo}	π_{algo}	ρ_{algo}	π_{algo}	ρ_{algo}
AN2C	0.90	97.48	0.85	94.67	0.81	94.92
AN2E	0.93	97.48	0.93	97.33	0.89	86.44
AR2	0.93	94.96	0.93	93.33	0.90	86.44
TR2M	0.91	94.12	0.90	93.33	0.83	91.53

Table 1: Efficiency and reliability statistics for OPM problems

Figure 1 and Table 1 suggest that the reliability of both AN2C and AN2E is a little better than that of AR2 and TR2M for all problem sizes. They also indicate that AN2C is somewhat slower iteration-wise than AR2 and TR2M, but AN2E is very comparable. The fact that the computationally more expensive AN2E is often faster than AN2C in terms of iteration numbers is not surprising. Indeed, the regularization term in (2.9) becomes $\sqrt{\sigma_k g_k}$ in convex regions, recovering the analysis of [33, 17], whereas AN2C regularizes the problem more strongly in (2.2) (by a factor 10 in our numerical settings) and therefore may further restricts the steplength. AN2E may however be computationally more intensive⁽²⁾ than AN2C. Which of the two algorithms is preferable in practice is likely to depend on the CPU cost of calculating the Hessian’s smallest eigenvalue.

As expected, the call (2.6) in AN2C is typically performed on very few iterations (for less 1.3% of them for the small-problems testset) and, when used, results in a negative-curvature step (2.12) even more exceptionally (as it turns out, never in our tests). This means in particular that a single linear-system solve was necessary for approximately 99% of all iterations. The AN2E variant of course called (2.6) at every iteration, but, again, (2.12) was never actually used. Our results also confirm the general effectiveness of a relatively simple implementation of the adaptive regularization algorithm AR2 using the test (5.1).

We also ran the SOAN2C and SOAN2E variants with $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-4}$, but their results are undistinguishable (for our test sets) from those obtained with AN2C and AN2E, except for a final eigenvalue analysis at the found approximate first-order point, which confirmed in

⁽²⁾Most failures of this algorithm on large problems occurred because the time limit was reached.

all cases that the second-order condition (4.1) did also hold at this point. No step of the form (4.4) was ever taken in our runs, despite the fact that such steps are necessary in theory (think of starting the minimization at a first-order saddle point).

These early results are encouraging but the authors are aware that only further experiments will allow a proper assessment of the method's true potential, both from the number of function/derivatives evaluations and CPU-usage points of view. Several further algorithmic developments within the new algorithms are also of interest, including the use of an iterative subproblem solver (allowed by our theory because of (2.3) and (2.10)), a possibly better balance between (2.2) and (2.6), approximate eigenvalue computations and the use of past information to speed them up, improved model decrease when a negative-curvature direction is computed, as well as refinements of the regularization parameter update (2.8), possibly in the spirit of [23].

6 Conclusions and Perspectives

We have proposed AN2C and AN2E, two second-order methods that alternate between Newton and negative-curvature directions for nonconvex problems. These methods differ from the more standard trust-region and adaptive-regularization techniques in that, beyond the required eigenvalue calculations, the involved step computation is free of further inner iterative processes and only requires the approximate solution of at most two (but typically one) linear systems per iteration. We have also proved that these algorithms require at most $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3/2})$ iterations to obtain an ϵ -approximate first-order critical point. Our proof builds on some elements of [33, 17] for the convex case and arguments for adaptive regularization [3] and other nonconvex optimization methods [15, 38]. At each iteration, the algorithms either take an explicit Newton step or negative curvature when it is sufficiently large compared to the square root of the gradient. The norm of the residuals of the Newton step are adjusted dynamically and different types of solvers can be used to solve the linear systems. The AN2C algorithm is constructed to limit the cost of evaluating eigenvalues as much as possible and require a single linear solve per iteration for a very large majority of problems. We also introduced algorithmic variants (SOAN2C and SOAN2E) which are guaranteed to find second-order critical points, and proved that they require at most $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3})$ iterations to do so. A first set of numerical experiments with the new methods indicates that they are very reliable and competitive with standard techniques in terms of number of iterations.

Promising lines of for future work include inexact derivatives, estimating the regularization parameter without evaluating the objective function (as in [24]), stochastic variants and the handling of simple constraints such as bounds on the variables in the spirit of [12, Section 14.2].

References

- [1] S. Bellavia, G. Gurioli, and B. Morini. Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. *IMA Journal of Numerical Analysis*, 41(1):764–799, 2021.
- [2] S. Bellavia, G. Gurioli, B. Morini, and Ph. L. Toint. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. *SIAM Journal on Optimization*, 29(4):2881–2915, 2019.
- [3] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming, Series A*, 163(1):359–368, 2017.

- [4] Y. Carmon and J. C. Duchi. Gradient descent efficiently finds the cubic-regularized non-convex Newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2021.
- [5] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In D. Precup and Y. W. Teh, editor, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 654–663. PLMR, 2017.
- [6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Trust-region and other regularization of linear least-squares problems. *BIT*, 49(1):21–53, 2009.
- [7] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. *SIAM Journal on Optimization*, 20(6):2833–2852, 2010.
- [8] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic overestimation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming, Series A*, 127(2):245–295, 2011.
- [9] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic overestimation methods for unconstrained optimization. Part II: worst-case function-evaluation complexity. *Mathematical Programming, Series A*, 130(2):295–319, 2011.
- [10] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In B. Sirakov, P. de Souza, and M. Viana, editors, *Invited Lectures, Proceedings of the 2018 International Conference of Mathematicians (ICM 2018)*, vol. 4, Rio de Janeiro, pages 3729–3768. World Scientific Publishing Co Pte Ltd, 2018.
- [11] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM Journal on Optimization*, 30(1):513–541, 2020.
- [12] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization*. Number 30 in MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, June 2022.
- [13] C. Cartis and K. Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming, Series A*, 159(2):337–375, 2018.
- [14] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 1 in MOS-SIAM Optimization Series. SIAM, Philadelphia, USA, 2000.
- [15] F. Curtis and D. Robinson. Exploiting negative curvature in deterministic and stochastic optimization. *Mathematical Programming, Series A*, 176(1-2):69–94, 2018.
- [16] F. Curtis, D. Robinson, C. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM Journal on Optimization*, 31(1):518–544, 2021.
- [17] N. Doikov and Yu. Nesterov. Gradient regularization of Newton method with Bregman distances. arXiv:2112.02952, 2021.
- [18] E. D. Dolan, J. J. Moré, and T. S. Munson. Optimality measures for performance profiles. *SIAM Journal on Optimization*, 16(3):891–909, 2006.
- [19] M. Ferris, S. Lucidi, and M. Roma. Nonmonotone curvilinear line search methods for unconstrained optimization. *Computational Optimization and Applications*, 6(2):117–136, 1996.
- [20] D. Goldfarb. The use of negative curvature in minimization algorithms. Technical Report TR80-412, Department of Computer Science, Cornell University, Ithaca, New York, USA, 1980.
- [21] S. M. Goldfeldt, R. E. Quandt, and H. F. Trotter. Maximization by quadratic hill-climbing. *Econometrica*, 34:541–551, 1966.
- [22] N. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Exploiting negative curvature in a linesearch algorithm for unconstrained optimization. *Optimization Methods and Software*, 14(1-2):75–98, 2000.
- [23] N. I. M. Gould, M. Porcelli, and Ph. L. Toint. Updating the regularization parameter in the adaptive cubic regularization algorithm. *Computational Optimization and Applications*, 53(1):1–22, 2012.
- [24] S. Gratton, S. Jerad, and Ph. L. Toint. Convergence properties of an objective-function-free optimization regularization algorithm, including an $\mathcal{O}(\epsilon^{-3/2})$ complexity bound. arXiv:2203.09947, 2022.
- [25] S. Gratton, S. Jerad, and Ph. L. Toint. First-order objective-function-free optimization algorithms and their complexity. arXiv:2203.01757, 2022.

- [26] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.
- [27] S. Gratton and Ph. L. Toint. OPM, a collection of optimization problems in Matlab. arXiv:2112.05636, 2021.
- [28] S. Gratton and Ph. L. Toint. Adaptive regularization minimization algorithms with non-smooth norms. *IMA Journal of Numerical Analysis*, (to appear), 2022.
- [29] A. Griewank. The modification of Newton’s method for unconstrained optimization by bounding cubic terms. Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, United Kingdom, 1981.
- [30] M. Liu, Z. Li, X. Wang, J. Yi, and T. Yang. Adaptive negative curvature descent with applications in non-convex optimization. In *Advances in Neural Information Processing Systems, vol. 31*. Curran Associates, Inc., 2018.
- [31] S. Lucidi, F. Rochetich, and M. Roma. Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization. *SIAM Journal on Optimization*, 8(4):916–939, 1998.
- [32] G. P. McCormick. A modification of Armijo’s step-size rule for negative curvature. *Mathematical Programming*, 13(1):111–115, 1977.
- [33] K. Mischenko. Regularized Newton method with global $\mathcal{O}(1/k^2)$ convergence. arXiv:2112.02089, 2021.
- [34] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Mathematical Programming*, 16(1):1–20, 1979.
- [35] Yu. Nesterov. *Lectures on Convex Optimization*. Springer Verlag, Heidelberg, Berlin, New York, 2018.
- [36] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming, Series A*, 108(1):177–205, 2006.
- [37] M. Porcelli and Ph. L. Toint. A note on using performance and data profiles for training algorithms. *ACM Transactions on Mathematical Software*, 45(2):1–25, 2019.
- [38] C. W. Royer and S. J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1448–1477, 2018.
- [39] K. Ueda and N. Yamashita. A regularized Newton method without line search for unconstrained optimization. *Computational Optimization and Applications*, 59(1-2):321–351, 2014.
- [40] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *Mathematical Programming, Series A*, 184((1-2)):35–70, 2020.
- [41] Z. Yao, P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Inexact non-convex Newton-type methods. *INFORMS Journal on Optimization*, 3(2):154–182, 2021.

A The test problems and their dimensions

Problem	n	Problem	n	Problem	n	Problem	n	Problem	n	Problem	n
argauss	3	chebyqad	10	dixmaanl	12	heart8ls	8	msqrtals	16	scurly10	10
arglina	10	cliff	2	dixon	10	helix	3	msqrtbls	16	scosine	10
arglinb	10	clplatea	16	dqartic	10	hilbert	10	morebv	12	sisser	2
arglinc	10	clplateb	16	edensch	10	himln3	2	nlminsurf	16	spmsqrt	10
argtrig	10	clustr	2	eg2	10	himm25	2	nondquar	10	tcontact	49
arwhead	10	cosine	10	eg2s	10	himm27	2	nzf1	13	tquartic	10
bard	3	crglvy	4	eigenals	12	himm28	2	osbornea	5	trigger	7
bdarwhd	10	cube	2	eigenbls	12	himm29	2	osborneb	11	tridia	10
beale	2	curly10	10	eigencls	12	himm30	3	penalty1	10	tlminsurf	16
biggs5	5	dixmaana	12	engvall	10	himm32	4	penalty2	10	tnlminsurf	16
biggs6	6	dixmaanb	12	engval2	3	himm33	2	penalty3	10	vardim	10
brownden	4	dixmaanc	12	expfit	2	hypcir	2	powellbs	2	vibrbeam	8
booth	2	dixmaand	12	extrosnb	10	indef	10	powellsg	12	watson	12
box3	3	dixmaane	12	fminsurf	16	integreq	10	powellsq	2	wmsqrtals	16
brkmcc	2	dixmaanf	12	freuroth	4	jensmp	2	powr	10	wmsqrtbls	16
brownal	10	dixmaang	12	genhumps	5	kowosb	4	recipe	2	woods	12
brownbs	2	dixmaanb	12	gottfr	2	lminsurf	16	rosenbr	10	yfitu	3
broyden3d	10	dixmaani	12	gulf	4	mancino	10	s308	2	zangwill2	2
broydenbd	10	dixmaanj	12	hairly	2	mexhat	2	sensors	10	zangwill3	3
chandheu	10	dixmaank	12	heart6ls	6	meyer3	3	schmvett	3		

Table 2: The OPM small test problems and their dimension

Problem	n	Problem	n	Problem	n	Problem	n	Problem	n	Problem	n
arglina	400	crglvy	400	dixmaanj	600	fminsurf	400	ncb20b	500	spmsqrt	997
arglinb	50	cube	500	dixmaank	600	freuroth	500	ncb20c	500	tcontact	400
arglinc	50	curly10	500	dixmaanl	600	helix	500	nlminsurf	400	tquartic	500
argtrig	50	deconvu	51	dixon	500	hilbert	500	nondquar	500	tridia	500
arwhead	500	dixmaana	600	dqartic	500	hydc20ls	99	nzf1	520	tlminsurf	400
bdarwhd	500	dixmaanb	600	edensch	500	indef	500	penalty1	500	tnlminsurf	400
brownal	500	dixmaanc	600	eg2	400	integreq	500	penalty2	100	vardim	500
broyden3d	500	dixmaand	600	eg2s	400	lminsurf	400	penalty3	500	wmsqrtals	400
broydenbd	500	dixmaane	600	eigenals	110	mancino	500	powellsg	500	wmsqrtbls	400
chandheu	500	dixmaanf	600	eigenbls	110	msqrtals	400	powr	500	woods	500
chebyqad	150	dixmaang	600	eigencls	110	msqrtbls	400	rosenbr	100		
clplatea	400	dixmaanb	600	engvall	500	morebv	500	sensors	100		
clplateb	400	dixmaani	600	extrosnb	500	ncb20	500	scosine	500		

Table 3: The OPM medium-size test problems and their dimension

Problem	n	Problem	n	Problem	n	Problem	n	Problem	n
arwhead	2000	dixmaand	2400	eg2	1600	integreq	2000	powellsg	2000
bdarwhd	2000	dixmaane	2400	eg2s	1600	lminsurf	4900	powr	2000
broyden3d	2000	dixmaanf	2400	eigenals	2550	msqrtals	1600	rosenbr	2000
broydenbd	2000	dixmaang	2400	eigenbls	2550	msqrtbls	1600	spmsqrt	1498
clplatea	4900	dixmaanb	2400	eigencls	2550	morebv	5000	tcontact	4900
clplateb	4800	dixmaani	2400	engvall	2000	ncb20b	2000	tquartic	2000
crglvy	4000	dixmaanj	2400	extrosnb	2000	ncb20c	2000	tridia	2000
cube	2000	dixmaank	2400	fminsurf	4900	nlminsurf	4900	tlminsurf	4900
curly10	1000	dixmaanl	2400	freuroth	2000	nondquar	2000	tnlminsurf	4900
dixmaana	2400	dixon	2000	helix	2000	nzf1	2600	vardim	2000
dixmaanb	2400	dqartic	2000	hilbert	2000	penalty1	2000	woods	2000
dixmaanc	2400	edensch	2000	indef	2000	penalty3	2000		

Table 4: The OPM largish test problems and their dimension